Certamen 2 ELO-329 Diseño y Programación Orientada a Objetos (1s2023)

En este certamen usted no podrá hacer preguntas. Si algo no está claro, indíquelo en su respuesta, haga una suposición razonable y resuelva conforme a ella.

Segunda parte, **con apuntes**:

Tercera pregunta (35 pts.) Simulación de un timbre luminoso

Un timbre luminoso es aquel que enciende una luz de la casa al ser presionado. En la simulación por software, el timbre es simulado por un botón y la luz es simulada por un rectángulo que cambia de color negro a amarillo.

A. (10 pts.) En Qt-creator cree el proyecto C2P3A con main.cpp y las clases necesarias para mostrar un único botón. Al presionar el botón, el programa imprime en consola "Timbre accionado". Vea aquí el resultado esperado. Suba a AULA un archivo como c2p3a.tar (o c2p3a.zip) que contenga los archivos .h, .cpp, .ui y .pro de su proyecto.

B. (10 pts.) En Qt-creator cree el proyecto C2P3B con main.cpp y las clases necesarias para mostrar un botón y un rectángulo negro (black). Al presionar el botón, el rectángulo debe cambiar a color amarillo (yellow). Vea aquí el resultado esperado. Suba a AULA un archivo como c2p3b.tar (o c2p3a.zip) que contenga los archivos .h, .cpp, .ui y .pro de su proyecto.

C. (15 pts.) En Qt-creator cree el proyecto C2P3C con main.cpp y las clases necesarias para mostrar un botón y un rectángulo negro. Cada vez que se presiona el botón, el rectángulo cambia a color amarillo por 2 segundos y vuelve a color negro. Vea aquí el resultado esperado. Suba a AULA un archivo como c2p3c.tar (o c2p3c.zip) que contenga los archivos .h, .cpp, .ui y .pro de su proyecto.

En los tres casos, usted debe considerar las ranuras (slots) que sean necesarias. En caso de usar connect(…), **debe** usar el prototipo de 4 parámetros visto en clases.

Usted puede acceder a la solución en:
http://profesores.elo.utfsm.cl/~agv/elo329/1s23/ELO329_1s23_C2P3_Sol/

A 10 pts.

// main.cpp

```
#include "mainwindow.h"
#include <QApplication>
int main(int argc, char *argv[]) // 2 pts método main
{
  QApplication a(argc, argv);
  MainWindow w;
  w.show();
  return a.exec();
}
```

//mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE
class MainWindow : public QMainWindow
{
Q_OBJECT
public:
 MainWindow(QWidget *parent = nullptr);
 ~MainWindow();
private slots:  // 2 pts
 void on_timbreBtn_clicked();
private:
 Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

// mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <iostream>
using namespace std;
MainWindow::MainWindow(QWidget *parent)
 : QMainWindow(parent)
 , ui(new Ui::MainWindow)
{
 ui->setupUi(this);
}
MainWindow::~MainWindow() {
 delete ui;
}
void MainWindow::on_timbreBtn_clicked()//2 pts
{
 cout << "Timbre accionado" << endl;
}
```

4 pts por presencia de botón en interfaz y correcta vinculación de la señal del botón con el slots de mainwindow vía módulo design o vía connect (aquí se muestra la solución vía módulo design)

B 10 pts.

```
#include "mainwindow.h"
#include <QApplication>
int main(int argc, char *argv[]) // 1 pts método main
{
 QApplication a(argc, argv);
 MainWindow w;
 w.show();
 return a.exec();
}
```

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
#include <QGraphicsRectItem>
#include <QGraphicsScene>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
 Q_OBJECT
public:
 MainWindow(QWidget *parent = nullptr);
 ~MainWindow();
```

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <iostream>
#include <QBrush>

MainWindow::MainWindow(QWidget *parent)
 : QMainWindow(parent)
 , ui(new Ui::MainWindow)
{   // 2 pts constructor
 ui->setupUi(this);
 ui->rectView->setScene(&scene);
 rectangle=scene.addRect(0,0,40,30);
 rectangle->setBrush(Qt::black);
}
MainWindow::~MainWindow()
{
```

```
private slots:          // 1 pts
  void on_timbreBtn_clicked();
private:
  Ui::MainWindow *ui;
  QGraphicsRectItem * rectangle; // 1 pts
  QGraphicsScene scene; //pudo estar en
                // constructor
};
#endif // MAINWINDOW_H
```

```
  delete ui;
}
void MainWindow::on_timbreBtn_clicked()//2 pts
{
  rectangle->setBrush(Qt::yellow);
}
```

3 pts por presencia de botón en interfaz y correcta vinculación de la señal del botón con el slots de mainwindow vía módulo design o vía connect (aquí se muestra la solución vía módulo design)

C 15 pts.

```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[]) // 1 pts
{
  QApplication a(argc, argv);
  MainWindow w;
  w.show();
  return a.exec();
}
```

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
#include <QGraphicsRectItem>
#include <QGraphicsScene>
#include <QTimer>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
  Q_OBJECT
public:
  MainWindow(QWidget *parent = nullptr);
  ~MainWindow();
private slots:     // 2 pts
  void on_timbreBtn_clicked();
  void myTimeout();

private:
  Ui::MainWindow *ui;
  QGraphicsRectItem * rectangle;
  QGraphicsScene scene;
```

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <iostream>
# include <QBrush>
#include <QObject>

MainWindow::MainWindow(QWidget *parent)
  : QMainWindow(parent)
  , ui(new Ui::MainWindow)
{
  ui->setupUi(this);
  ui->rectView->setScene(&scene);
  rectangle = new QGraphicsRectItem(0,0, 40,30);
  scene.addItem(rectangle);
  rectangle->setBrush(Qt::black);//hasta aquí 1pts
  connect(&myTimer,SIGNAL(timeout()), this,  SLOT(myTimeout()));  //
4 pts
}
MainWindow::~MainWindow()
{
  delete ui;
}
void MainWindow::myTimeout() { // 2 pts
  rectangle -> setBrush(Qt::black);
  myTimer.stop();
```

```
  QTimer myTimer;  // 1 pts
};
#endif // MAINWINDOW_H
```

```
}
void MainWindow::on_timbreBtn_clicked() // 2 pts
{
  myTimer.start(2000);
  rectangle->setBrush(Qt::yellow);
}
```

2 pts por presencia de botón en interfaz y correcta vinculación de la señal del botón con el slots de mainwindow vía módulo design o vía connect (aquí se muestra la solución vía módulo design)